ДЕТСКАЯ ОНЛАЙН-ШКОЛА ПРОГРАММИРОВАНИЯ
HELLO WORLD

#ШПАРГАЛОЧКИ

СОЗДАНИЕ САЙТОВ FRONT-END РАЗРАБОТКА

Дополнительный уровень

Материалы подготовлены отделом методической разработки

Больше полезных материалов и общения в нашем комьюнити в Telegram: https://t.me/hw_school

FLUX архитектура (часть 1)

Перерисовка страницы

После того как сообщение было добавлено, страницу надо отрисовать заново. За отрисовку отвечает index.js, а именно - функция **ReactDOM.render()**. Поскольку теперь отрисовка нужна после каждого добавленного сообщения, нужно создать отдельную функцию, а вызов **ReactDOM.render** переместить внутрь нее.

Но оставить эту функцию в index.js и просто вызывать ее после добавления сообщения нельзя, так как это вызовет **циклическую зависимость**.

Перерисовка страницы

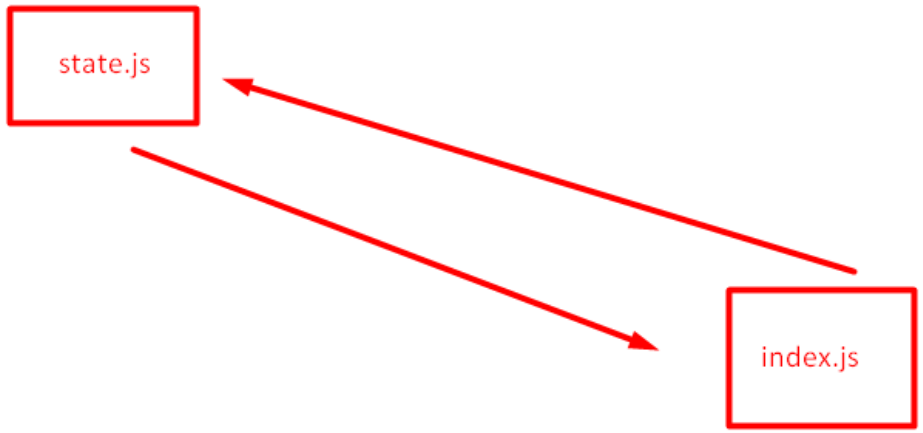
```
export let rerenderTree = () => {

        ReactDOM.render(

                <React.StrictMode>

                        <App state={state} addPost={addPost} sendMessage={sendMessage} />

                </React.StrictMode>,

                document.getElementById('root')

        );

}
```

Циклическая зависимость

Циклическая зависимость - это зависимость двух участков кода друг от друга. Например, если перерисовывающую страницу функцию из **index.js** вызывать в сохраняющей сообщение функции из **state.js**, возникнет циклическая зависимость между этими двумя файлами.

Иными словами, **index.js** импортирует **state**, а **state.js** импортирует функцию отрисовки из **index.js**.

Циклическая зависимость

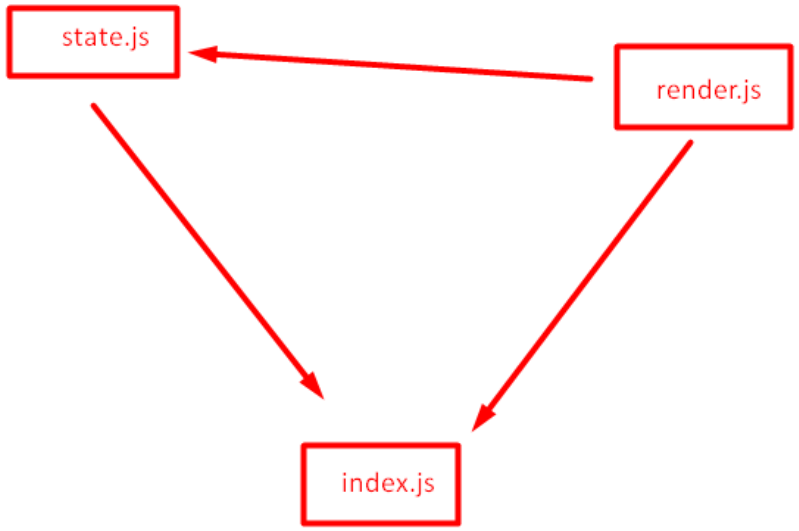


Циклическая зависимость

Чтобы избавиться от циклической зависимости, **весь код из index.js** нужно вынести в отдельный файл - например, **render.js** (он может храниться в папке **src**).

Тогда **index.js** будет только импортировать и вызывать функцию, отрисовывающую страницу. А в **state.js** изменится только место, откуда эта функция импортируется.

Циклическая зависимость



unshift

unshift - метод, добавляющий элемент в массив. Используется так же, как push, но добавляет новый элемент в **начало массива**, а не его конец.

```
state.profilePage.postsData.unshift(newPost)

// теперь новые посты добавляются в начало массива и отрисовываются первыми, а не последними
```